LaTeX en quatre pages

Pierre Collet
LSIIT — Université de Strasbourg

Ce document montre l'utilisation d'une sélection de commandes LaTeX qui devraient permettre de réaliser pratiquement n'importe quel document (comme un rapport de projet, par exemple). Le fichier source fourni permet de voir comment les choses sont faites, même s'il est un peu compliqué, car intégrant les commandes à utiliser (mises en page à l'aide de la commande `\verb` et de l'environnement `verbatim`). Pour en savoir plus, Google est votre ami.

1 Création du fichier

Un document LaTeX n'est rien d'autre qu'un fichier *texte* en ascii standard. Le meilleur moyen de le créer est donc d'utiliser un éditeur de textes comme `vi` ou `emacs`. Traditionnellement, le nom du fichier source est suivi de l'extension `.tex` (`initLaTeX.tex` pour ce fichier).

Pour la compilation, LaTeX est fourni en standard dans les distributions Linux. Sur Windaube, je conseille de télécharger MiKTeX (`http://miktex.org/`) car ça s'installe sans effort.

2 LaTeX, mode d'emploi

2.1 Latexification

Une fois votre fichier créé, ouvrez un terminal (fenêtre de commande sur windaube, terminal sous linux), allez dans le bon répertoire et tapez `latex monfichier.tex` pour obtenir un fichier `.dvi`, ou `pdflatex monfichier.tex` pour obtenir un fichier `.pdf`.

Si tout se passe bien, pas mal de choses défileront à l'écran qui se termineront par :

```
...
Output written on xxx.dvi (X pages, XX bytes)
Transcript written on xxx.log       XX seconds
```

Cela vous indique qu'un fichier `xxx.dvi` vient d'être créé. Vous pouvez passer à la phase suivante (visualisation ou impression avec `xdvi`, `kdvi` ou `yap`, *cf.* section 2.2).

En cas de problème, le message sera du type :

```
...
!Undefined control sequence
l.189 bla bla bla bla bla bla
                              bla bla bla bla bla bla
?
```

Cela signifie qu'il y a une erreur que LaTeX situe à la ligne 189 du fichier source, au niveau de la cassure dans le texte. Si vous avez compris de quoi il s'agit, tapez "x" pour sortir, puis modifiez votre texte en conséquence avec votre éditeur préféré.

LaTeX est un langage compilé et les erreurs sont donc similaires à celles qu'on peut avoir si l'on compile un fichier C, du style, si un $ (qui permet de passer en "mode mathématique" n'est pas refermé après la formule mathématique, l'erreur apparaîtra forcément plus loin (car le compilateur ne peut pas savoir où il fallait arrêter la formule).

2.2 Visualisation et impression

Le fichier créé par le compilateur est un `.dvi`, pour *DeVice Independent*, c'est-à-dire que c'est un fichier "vectoriel," indépendant de la résolution finale de la machine permettant de visualiser le document (écran, imprimante, ...). Cela permet au fichier `dvi` d'être tout petit (une dizaine de k-octets pour ce document de 4 pages, à comparer avec 4 pages écrites en Mot, de PetitMou contenant des formules mathématiques). Les fichiers `.dvi` sont donc visualisables sur tout. Ils sont composés de caractères ASCII standards (entre 32 et 128) ce qui leur garantit d'être portables sur *toute* machine, indépendemment du jeu de caractères étendus utilisé.

Pour plus de compatibilité, il est possible de transformer un fichier `dvi` en postscript (langage vectoriel aussi) avec la commande `dvips` (disponible sur tout système d'exploitation). Ensuite, si vous voulez du pdf, tout fichier `ps` est transformable en `pdf` par la commande `ps2pdf`.

Pour voir votre œuvre en économisant du papier, le visualisateur de fichier `dvi` est `xdvi` (ou plus récemment `kdvi`) sous Linux, et `yap` avec package MiKTeX sous Windaube. Si vous avez créé un `pdf` directement avec `pdflatex`, alors, vous savez probablement comment l'ouvrir.

3 LaTeX, principes généraux

Voici un entête (trop) complet permettant de maîtriser la mise en pages d'un document :

```
\documentclass{article}          % pour un document du style "article"
                                 % (mais il y a "book", "letter", ...)
\usepackage[francais]{babel}     % pour mise en page française, césure, ...
\usepackage[utf8]{inputenc}      % ligne à utiliser pour ascii étendu unix
%\usepackage[T1]{fontenc}        % ligne à utiliser pour ascii étendu windaube
\usepackage{graphicx}            % pour insérer des images

\righthyphenmin 3                % Nb de caract. min à laisser en fin de ligne
\lefthyphenmin 3                 % (resp.) en début de ligne lors d'une césure
%\parindent 5mm                  % pour forcer la largeur de l'indentation
\flushbottom                     % pour que les pages terminent bien en bas
\parskip 6pt plus 2pt minus 2pt  % séparation variable des paragraphes (pour
                                 % que \flushbottom puisse fonctionner)
\voffset -1cm                    % décalage hard sur la page
\topmargin 0cm                   % marge en haut de page
\textheight 22cm                 % hauteur du texte
\oddsidemargin 1cm               % marges pages impaires
\evensidemargin 1cm              % marges pages paires
\textwidth 14.5cm                % largeur du texte

\begin{document}
% Vous pouvez taper votre texte ici. Comme vous vous en doutez, tout ce qui
% suit un % est ignoré.
\end{document}
```

La plupart de ces options peuvent être omises, et ne sont signalées que pour permettre à l'utilisateur de modifier les styles parfois un peu stricts de LaTeX (mais qui correspondent aux règles de bonne typographie).

Lorsque le style `article` est utilisé, un certain nombre de champs sont prédéfinis permettant à LaTeX de gérer tout tout seul. Par exemple, en informant LaTeX du titre, de l'auteur de l'article et de la date, il est possible de demander une mise en page automatique avec `\maketitle`.

```
\title{\LaTeX\ mode d'emploi}   % si la date est omise, la date du jour de la
\author{Pierre Collet}          % compilation sera utilisée
\date{19/11/09}                 % title, author, date sont des variables
\maketitle                      % utilisées par la macro \maketitle
```

LaTeX ne tient compte ni du nombre d'espaces insérés entre deux mots ni d'un passage à la ligne (un terminal "standard" affiche 80 colonnes). Il faut donc sauter une ligne blanche pour démarrer un nouveau paragraphe. Ensuite, les différentes sections du texte s'obtiennent avec :

```
\section{Une section}           \subsubsection{Une sous-sous-section}
\subsection{Une sous-section}   \paragraph{Un paragraphe}
```

Elles sont numérotées (sauf les paragraphes) et mises en page automatiquement.

4 Enrichissements, caractères spéciaux et environnements

Enrichissements Les principaux enrichissements du texte sont les suivants :
`{\it` *italiques*`}`, `{\em` *emphasized*`}`, `{\sl` *slanted*`}`, `{\tt teletype}`, `{\sc` SMALL CAPITALS`}`, `{\bf` **bold face**`}`, `{\large` grand`}`, `{\Large` plus grand`}`, `{\LARGE` encore plus grand`}`, `{\huge` himmense`}`, `{\small` petit`}`, `{\footnotesize` plus petit`}`, `{\scriptsize` encore plus petit`}`, `{\tiny` minuscule`}`.

On peut composer `\bf` et `\em` pour obtenir `{\bf\em` ***du texte gras en italique***`}`.

`{\em` *L'enrichissement "emphasize" est* `{\em` intelligent`}` *au sens où comme on est en train de le voir, un second* `{\em` remet le texte en roman`}` *pour le distinguer du texte en italiques*`}`. *C'est la différence avec* `{\it` *qui force l'utilisation d'italiques*`}` *même si on est déjà en italiques*`}`.

Caractères spéciaux Les caractères #, $, %, &, _, {, } étant spéciaux, il faut les faire précéder d'un `\` pour les obtenir. On peut insérer une espace avec `\` suivi d'un espace. Les guillemets s'obtiennent avec ` `` ` pour " et `''` pour ". `\ldots` produit "..." et `$\backslash$` donne \.

Pour | encadrer un texte tenant sur une ligne, | taper `\fbox{encadrer un texte}`[1].

\vspace{1cm} laissera un espace vertical d'1cm, et \hspace{5mm} laissera un espace horizontal de 5mm entre deux mots. \newpage "force" un saut de page, \\ force un retour à la ligne et \noindent supprime l'indentation. \fbox{\vbox{...}} encadre plusieurs lignes.

Plusieurs environnements existent : `center`, `itemize`, `enumerate`, `description`,... :
`\begin{center}`

Ils s'utilisent de cette manière.

`\end{center}`

`\begin{itemize}`
- Une vache. (que l'on tape `\item Une vache.`)
- Un éléphant. (que l'on tape `\item Un éléphant.`)

`\end{itemize}`

Si l'on remplace `itemize` par `enumerate` sans rien changer aux items, on obtient :

1. Une vache.
2. Un éléphant.

[1] et `\footnote{pour une note de bas de page}`.

L'environnement `description` permet de décrire facilement une suite d'objets :

Mot Traitement de textes buggué, vendu très cher par PetitMou, mais que tout le monde s'arrache grâce à des techniques de marketing intelligentes (consistant notamment à sortir périodiquement des versions incompatibles entre elles, *cf.* nouveau format `.docx`).

LaTeX Traitement de textes sans bugs et gratuit, mais inconnu car ne bénéficiant pas des techniques de marketing intelligentes décrites ci-dessus (la première version de ce document a été écrite en LaTeX en 1990 et est toujours compilable en 2010 !).

À noter que la documentation papier de Word 3.0 était écrite en LaTeX, ce qui en dit long sur la capacité de Mot à faire ce pour quoi il est vendu ...

4.1 Commandes un peu plus avancées

Le tableau

justifié à gauche	texte centré	justifié à droite
left	center	right

s'obtiendra en écrivant :

```
\begin{tabular}{|l||cr|}
\hline
justifié à gauche & texte centré & justifié à droite\\ \hline\hline
left & center & right\\
\end{tabular}
```

Les options entre accolades après `\begin{tabular}` permettent de décrire la structure du tableau. Les barres verticales symbolisent les lignes verticales séparant les colonnes. On voit qu'il y a trois colonnes définies, une justifiée à gauche (`l`), la deuxième est centrée (`c`), et la troisième justifiée à droite (`r`). Les lignes horizontales sont obtenues par des `hline` ajoutées après le retour à la ligne (obtenu avec `\\`, à ne pas oublier en fin de ligne). Dans le tableau, le contenu des cases est séparé par le signe &.

On peut doubler les lignes en doublant les `\hline` ou les `|` comme ci-dessus. L'absence de `\hline` ou de `|` résulte en l'absence de la ligne correspondante.

5 Écriture des mathématiques

C'est un réel plaisir, car il n'y a qu'à écrire la formule en LaTeX, encadrée par des $. Ainsi, `$\sqrt[5]{\sum_{x=0}^{n}\frac{1}{x^2}}$` donnera : $\sqrt[5]{\sum_{x=0}^{n}\frac{1}{x^2}}$.

Pour obtenir la même formule sur une ligne indépendante (et en moins "tassé" car il n'y a plus besoin de la faire tenir sur la hauteur d'une ligne de texte), il faut encadrer avec `\[...\]`.

Voici la même formule avec crochets : `\[\sqrt[5]{\sum_{x=0}^{n}\frac{1}{x^2}}\]`

$$\sqrt[5]{\sum_{x=0}^{n}\frac{1}{x^2}}$$

Notez la différence de composition (bornes de la somme, racine cinquième, ...).

6 Figures

Insérer une image se fera avec `\includegraphics[width=0.8\textwidth]{image.jpg}` ou `\includegraphics{image.eps}`. Attention : une image `ps` étant vectorielle, le compilateur `pdflatex` la recrachera. Inversement, une image `jpg` n'étant *pas* vectorielle, le compilateur `latex` standard la recrachera. Si l'on n'a que des figures vectorielles (`ps` par exemple), alors, on pourra latexifier normalement et à la fin seulement, obtenir un gros pdf en passant successivement par les moulinettes `dvips` pour obtenir un `.ps` suivi de `ps2pdf` pour avoir le `.pdf`.